Guideline

Guidelines when working with Active Forms and IBM Cognos TM1

Product(s): IBM Cognos TM1 10.2

Area of Interest: Financial Management

Copyright and Trademarks

While every attempt has been made to ensure that the information in this document is accurate and complete, some typographical errors or technical inaccuracies may exist. IBM does not accept responsibility for any kind of loss resulting from the use of information contained in this document. The information contained in this document is subject to change without notice.

This document is maintained by the IBM Business Analytics Proven Practices team. You can send comments, suggestions, and additions to cscogpp@ca.ibm.com.

Table of Contents

Introduction

Purpose of Document

This document will provide guidelines on how to create active forms for IBM Cognos TM1.

Applicability

IBM Cognos TM1 10.2

Exclusions and Exceptions

There are no known exceptions and exclusions at the time this document was created

Assumptions

The document assumes the user is familiar with IBM Cognos TM1 and that includes dimensionality, cubes, calculations, etc. The documentation will refer to dimensions and cubes, but will not provide any details on how to create those dimensions or cubes.

The document assumes that user is familiar with DBRW, which is an IBM Cognos TM1 worksheet function that is only valid for IBM Cognos TM1 Worksheets. Users may want to refer to the user guide section that discusses worksheet functions, including DBRW.

http://publib.boulder.ibm.com/infocenter/ctm1/v9r4m1/index.jsp?topic=/com.ibm.swg.im.cognos.tm1_ref.9.4.1.doc/tm1_ref_id12154wrk_dbrw.html

The document also requires user to be familiar with MDX statements. Please refer to the user guide documentation for more details on MDX statements

http://pic.dhe.ibm.com/infocenter/ctm1/v10r1m0/index.jsp?topic=%2Fcom.ibm.swg.ba.cognos.tm1_op.10.1.0.doc%2Fc_hierarchyproperties_n8547.html

Guidelines on how to create active forms with IBM Cognos TM1

What are active forms?

Active Forms let users view and update live IBM Cognos TM1 Cube data directly in Excel whenever users are connected to the server on which the cube data resides. Active Forms retain the ability to expand and collapse row dimension consolidations in IBM Cognos TM1 views while allowing you to use native Excel features and functions to create complex reports.

With the introduction of Active Forms, the dynamic slice functionality that was available in previous releases is no longer supported. If you open a worksheet containing a dynamic slice in IBM Cognos TM1 9.4, the slice is rendered as a 'classic' slice reflecting the state of the slice the last time it was saved. Any formatting applied to the dynamic slice is lost when the slice is opened in IBM Cognos TM1 9.4 or later.

Dynamic Row Content

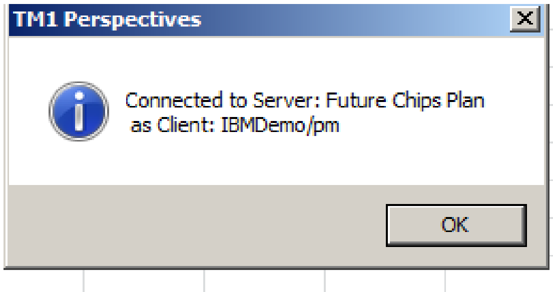
- User Defined Row Content
- Dynamic, in sheet formatting
- Drillable – up/down
- Zero Suppression
- Self-Management of Dimension members
- Excel Design -> Web Deployment

Benefits of Active Forms

- Enterprise-enables TM1Web
- Enhanced Maintainability – 1 Active Form provides 'personalized' deployment for 1000's
- Enhanced Application Design
    - Combine with Action Button to immediately plan for new members
    - Mix and Match Active Form slices
    - Contextual Flow between Active Forms
- Visually Enhanced Reports
    - 'Fast Start' Reporting
    - Conditional Excel Formatting (levels, Consol/Leaf, Expanded/Contracted)

How do you access the active forms?

Users are required to signed on to Microsoft Excel activate the connection to IBM Cognos TM1 server. Figure 1 shows the users is connected to IBM Cognos TM1 server under PM log in.


TM1 Perspectives

Connected to Server: Future Chips Plan
as Client: IBMDemo/pm

Figure 1 – shows users connected to TM1 Perspectives

Users will then have many options to choose from as excel is now connected to IBM Cognos TM1 server. The focus of the following steps will be on active forms and how defined the parameters for the active forms.

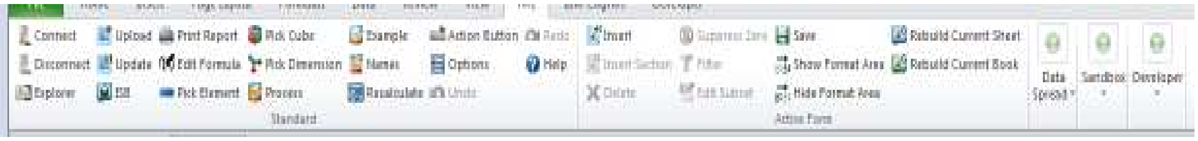
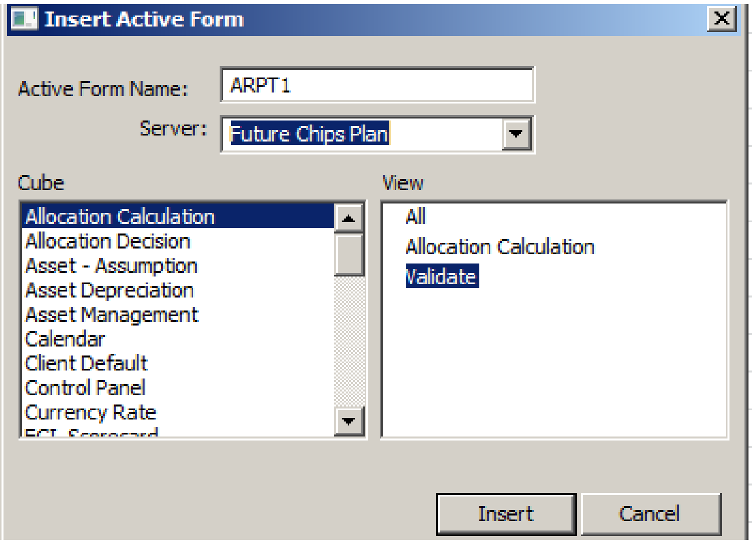


Figure 2 – shows menu options available within Excel for TM1

Users can then choose the cube to use within the active forms. The below example illustrates using the IBM Cognos TM1 Server called Future Chips along with the Allocation Cube. If the cube has views then the views will be available to users as part of the active form creation. The below example is using the validate view as part of the active form creation.


Insert Active Form

Active Form Name: ARPT1

Server: Future Chips Plan

Cube

- Allocation Calculation
- Allocation Decision
- Asset - Assumption
- Asset Depreciation
- Asset Management
- Calendar
- Client Default
- Control Panel
- Currency Rate
- ECI Scorecard

View

- All
- Allocation Calculation
- Validate

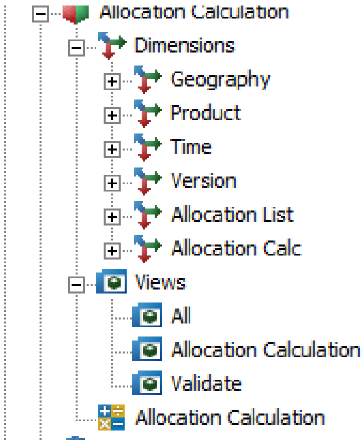
Figure 3 – shows insertion of an active form

Access from Cube Viewer

- Be careful to have a subset definition in the rows (particularly if using a large row definition)
    - Note: drilling in a view will remove the subset definition from status
- If you don't have the subset definition in the rows then the row values are sliced into a hidden sheet and the TM1RPTRow will show a reference to it: ex. ({AR}01'!$B$17:$B$22)
    - Sheet references will not perform as well
    - If not using, delete hidden sheets

Active Form example

It is good practice when working with IBM Cognos TM1 active forms to also have IBM Cognos TM1 architect open, especially the cube that users plan to include in the active form. Figure 4 will show architect and allocation calculation cube. Users will notice in figure 4 that you can see the dimensions, views, rules, etc.



- Allocation Calculation
  - Dimensions
    - Geography
    - Product
    - Time
    - Version
    - Allocation List
    - Allocation Calc
  - Views
    - All
    - Allocation Calculation
    - Validate
  - Allocation Calculation

Figure 4 – shows the allocation calculation cube structure

Active Form Functions

The IBM Cognos TM1 active form functions include:

- TM1RptView
  - Specifies
    - a name for the view
    - zero suppression for rows
    - which dimensions of the cube are used as title dimensions
    - a range that contains format information for the form
    - filter the view's column dimensions
      - TM1RptView(ViewID,ZeroSuppression,TM1RptTitle,TM1RPTFMTRNG,TM1RPTFMTIDCOL)

- Note: Because the TM1RptView determines the title dimensions (via TM1RptTitle parameters), a refresh of the active form will always filter the results based on the corresponding dimension elements. If for example a first iteration of the active form uses 'Actuals' as a TM1RptTitle filter for the version dimension, you cannot add a column with 'Forecast' values to the Active Form without removing the TM1RptView filter for 'Actuals': If you add a DBRW formula to the Active Form columns to also (in addition to 'Actuals') query 'Forecast' data & without making adjustments to the TM1RptTitle filters in TM1RptView, you may see the 'Forecast' values when refreshing the data (via pressing F9 for example), but a refresh of the Active Form will not query the 'Forecast' data. This is because as per TM1RptTitle filter, the Active Form will filter out only 'Actuals'. It follows that if an active form is enhanced or redesigned to include additional columns/data based and if this additional data requires querying more than one element from a dimension, the TM1RptTitle filter for this dimension needs to be removed from the TM1RptView formula.

- TM1RptTitle
    - Defines an Active Form title dimension.
        - Syntax: TM1RptTitle(Dimension,Element)

- TM1RptRow
    - Defining Report Rows
        - based on named ranges
        - based on a named subsets
        - MDX code directly on the spreadsheet
            - TM1RptRow(ReportView, Dimension, Subset, SubsetElements, Alias, ExpandAbove, MDXStatement, Indentations, ConsolidationDrilling)
                - **ReportView**: A reference to a cell that contains a TM1RptView formula.
                - **Dimension**: Specifies the TM1 server and the dimension in the form.
                - **Subset**: A named subset if it exists otherwise all dimensions are included.
                - **SubsetElements**: A cell range reference that specifies a list of elements. If not empty, then the Subset argument is ignored. If empty, then the elements from the subset specified by the Subset argument are used.
                - **Alias**: Defines the alias used for the subset.
                - **ExpandAbove**: Flag to turn on or off the subset Expand Above property where 1 = consolidated elements expand upward when drilling, and 0 =consolidated elements expand downward when drilling.

            - **MDXStatement**: An MDX statement that applies to the subset specified by the Subset argument. Indentations: An integer value to indicate how many indentations are applied to each level when drilling down on a consolidated element. If the argument value is 0, no auto-indentation is performed. When the value is missing, one indentation is applied to each level as you drill down on a consolidated element.

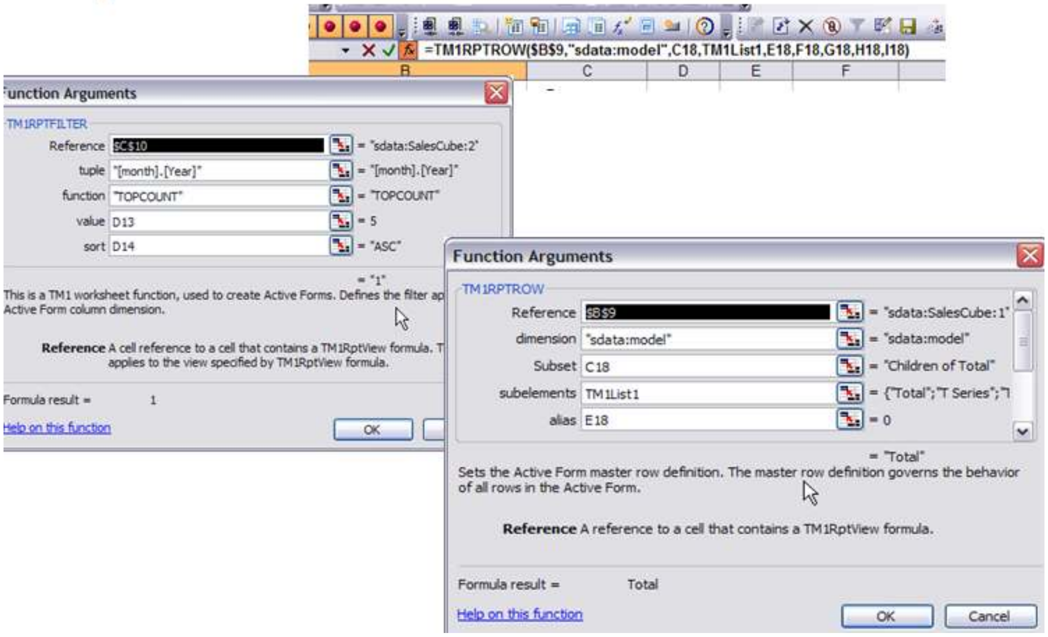
- TM1RptFilter
    - Defines the filter applied to an Active Form column dimension.
    - Syntax: TM1RptFilter(ReportView,Tuple,FilterFunction,FilterValue,SortOrder)

- TM1RptELev
    - Returns an integer value for an element level relative to root in the subset. This function is distinct from the ElLev worksheet function.
        - Syntax TM1RptElLev(RptRowFormula, Element)

- TM1RptEIsExpanded
    - Returns a boolean value to indicate whether an element is expanded in a row subset within an Active Form.
        - Syntax TM1RptElIsExpanded(RptRowFormula, Element)

- TM1RptEIsConsolidated
    - Returns a Boolean value to indicate whether an element in an Active Form is consolidated.
        - Syntax TM1RptElIsConsolidated(RptRowFormula, Element)

The attached link will provide additional information: http://publib.boulder.ibm.com/infocenter/ctm1/v9r4m1/index.jsp?topic=/com.ibm.swg.im.cognos.tm1_ref.9.4.1.doc/tm1_ref_id13340wrk_tm1rptrow.html



=TM1RPTROW($B$9,"sdata:model",C18,TM1List1,E18,F18,G18,H18,I18)

Function Arguments

TM1RPTFILTER

Reference $C$10 = "sdata:SalesCube:2"
tuple "[month].[Year]" = "[month].[Year]"
function "TOPCOUNT" = "TOPCOUNT"
value D13 = 5
sort D14 = "ASC"

= "1"

This is a TM1 worksheet function, used to create Active Forms. Defines the filter ap
Active Form column dimension.

Reference A cell reference to a cell that contains a TM1RptView formula. T
applies to the view specified by TM1RptView formula.

Formula result = 1

Function Arguments

TM1RPTROW

Reference $B$9 = "sdata:SalesCube:1"
dimension "sdata:model" = "sdata:model"
Subset C18 = "Children of Total"
subelements TM1List1 = {"Total";"T Series";"T
alias E18 = 0

= "Total"

Sets the Active Form master row definition. The master row definition governs the behavior of all rows in the Active Form.

Reference A reference to a cell that contains a TM1RptView formula.

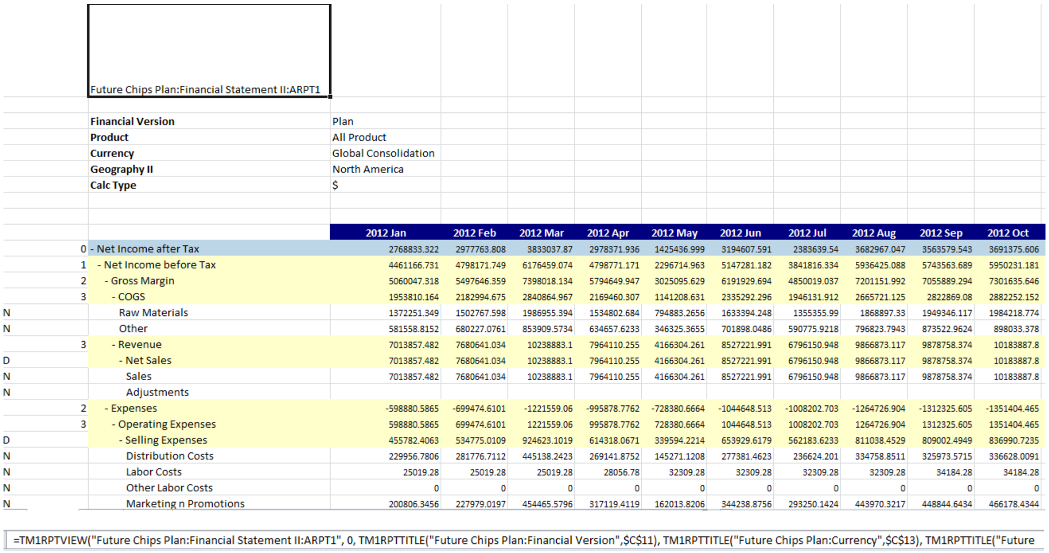
Formula result = Total

Figure 5 shows the defining of the TM1RPTFILTER and TM1RPTROW

Features Active Forms Support

- selectable title dimensions
- stacked row and column dimensions
- expandable/collapsible consolidations (rows only)
- suppress zeros on rows based on title dimension settings
- drill-through
- filtering
- data spreading

Figure 6 shows an active form for the Financial Statement. The report starts at C11 as defined by the formula TM1RPTitle.



Future Chips Plan:Financial Statement II:ARPT1

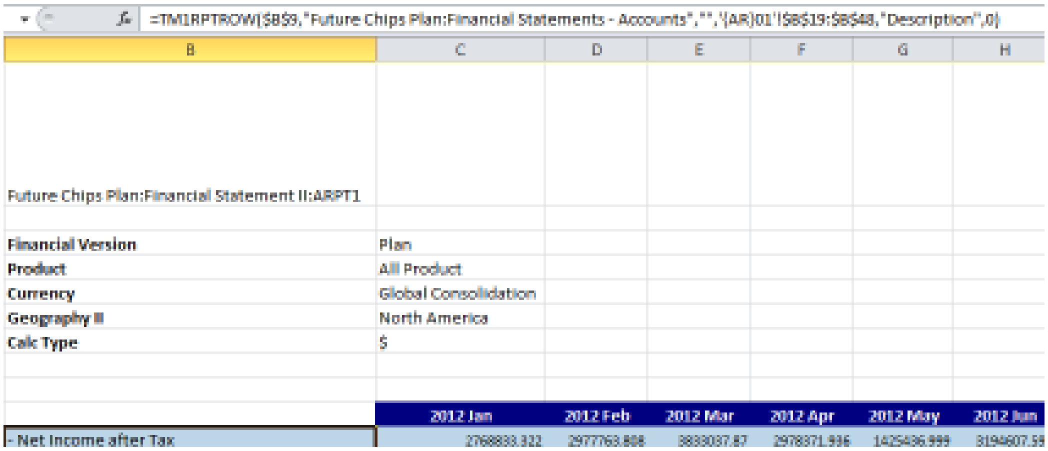
Financial Version	Plan
Product	All Product
Currency	Global Consolidation
Geography II	North America
Calc Type	$

			2012 Jan	2012 Feb	2012 Mar	2012 Apr	2012 May	2012 Jun	2012 Jul	2012 Aug	2012 Sep	2012 Oct
	0	- Net Income after Tax	2768833.322	2977763.808	3833037.87	2978371.936	1425436.999	3194607.591	2383639.54	3682967.047	3563579.543	3691375.606
	1	- Net Income before Tax	4461166.731	4798171.749	6176459.074	4798771.171	2296714.963	5147281.182	3841816.334	5936425.088	5743563.689	5950231.181
	2	- Gross Margin	5060047.318	5497646.359	7398018.134	5794649.947	3025095.629	6191929.694	4850019.037	7201151.992	7055889.294	7301635.646
	3	- COGS	1953810.164	2182994.675	2840864.967	2169460.307	1141208.631	2335292.296	1946131.912	2665721.125	2822869.08	2882252.152
N		Raw Materials	1372251.349	1502767.598	1986955.394	1534802.684	794883.2656	1633394.248	1355355.99	1868897.33	1949346.117	1984218.774
N		Other	581558.8152	680227.0761	853909.5734	634657.6233	346325.3655	701898.0486	590775.9218	796823.7943	873522.9624	898033.378
	3	- Revenue	7013857.482	7680641.034	10238883.1	7964110.255	4166304.261	8527221.991	6796150.948	9866873.117	9878758.374	10183887.8
D		- Net Sales	7013857.482	7680641.034	10238883.1	7964110.255	4166304.261	8527221.991	6796150.948	9866873.117	9878758.374	10183887.8
N		Sales	7013857.482	7680641.034	10238883.1	7964110.255	4166304.261	8527221.991	6796150.948	9866873.117	9878758.374	10183887.8
N		Adjustments										
	2	- Expenses	-598880.5865	-699474.6101	-1221559.06	-995878.7762	-728380.6664	-1044648.513	-1008202.703	-1264726.904	-1312325.605	-1351404.465
	3	- Operating Expenses	598880.5865	699474.6101	1221559.06	995878.7762	728380.6664	1044648.513	1008202.703	1264726.904	1312325.605	1351404.465
D		- Selling Expenses	455782.4063	534775.0109	924623.1019	614318.0671	339594.2214	653929.6179	562183.6233	811038.4529	809002.4949	836990.7235
N		Distribution Costs	229956.7806	281776.7112	445138.2423	269141.8752	145271.1208	277381.4623	236624.201	334758.8511	325973.5715	336628.0091
N		Labor Costs	25019.28	25019.28	25019.28	28056.78	32309.28	32309.28	32309.28	32309.28	34184.28	34184.28
N		Other Labor Costs	0	0	0	0	0	0	0	0	0	0
N		Marketing n Promotions	200806.3456	227979.0197	454465.5796	317119.4119	162013.8206	344238.8756	293250.1424	443970.3217	448844.6434	466178.4344

```
=TM1RPTVIEW("Future Chips Plan:Financial Statement II:ARPT1", 0, TM1RPTTITLE("Future Chips Plan:Financial Version",$C$11), TM1RPTTITLE("Future Chips Plan:Currency",$C$13), TM1RPTTITLE("Future
```

Figure 6 – shows budget financial Statements

Review the net income before taxes and the formula will reference the row and IBM Cognos TM1 Cube source. The formula is specifying the IBM Cognos TM1 server (Future Chips) and the IBM Cognos TM1 Cube (Financial Statements – Accounts) as the source.



```
=TM1RPTROW($B$9,"Future Chips Plan:Financial Statements - Accounts","",'{AR}01'!$B$19:$B$48,"Description",0)
```

B	C	D	E	F	G	H
Future Chips Plan:Financial Statement II:ARPT1						
Financial Version	Plan					
Product	All Product					
Currency	Global Consolidation					
Geography II	North America					
Calc Type	$					
	2012 Jan	2012 Feb	2012 Mar	2012 Apr	2012 May	2012 Jun
- Net Income after Tax	2768833.322	2977763.808	3833037.87	2978371.936	1425436.999	3194607.59

Figure 7 – shows Net Income reference

-NB-

- Users can add new dimensions to the form, but the new dimensions must be valid. The recommended practice is to have TM1 Architect open as you review the cubes/dimensions to ensure the proper names are brought into the active form.
- Users also need to be familiar with the correct syntax when using the formulas in the spreadsheet.
- Zero suppressions is "1" means its on and "0" means zero suppression is off

Active Form Design

- Use Classic Slice if row set is fixed or blend Classic and Active Form sections
- Unnamed subsets in Row dimension(s) will create a hidden sheet with element contents
    - Use named subsets
    - Use MDX statement in TM1RPTROW formula (very powerful)
        - =TM1RPTROW($C$9,"bpm:Products","","","bob",,Q14)
        - Cell Q14 contains ….{TM1DRILLDOWNMEMBER({[Products].[Total]}, ALL, RECURSIVE)}
- View statement for additional data retrieval
- Formatting Best Practices apply (and can be even more important due to the dynamic nature of the row count). Please refer to the customer support site for list of support functions: https://www-304.ibm.com/support/docview.wss?uid=swg27027528

Support and use of MDX formulas

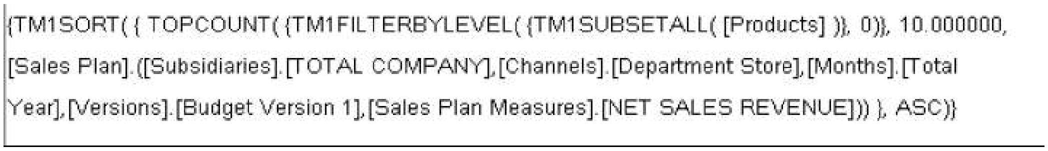
IBM Cognos TM1 supports several IBM Cognos TM1 specific MDX expressions. You can apply these expressions while developing MDX applications to run against the server or when creating/editing dynamic subsets in the Expression Window of the Subset Editor.

- TM1FILTERBYPATTERN(<set>, <pattern_str>)
  This TM1 -specific MDX function returns all the members in <*set*> with names matching the pattern <*pattern_str*>.
- TM1FILTERBYLEVEL(<set>, <level_number>)
  This TM1 -specific MDX function returns all the members in <*set*> of the specified <*level_number*>.
- TM1DRILLDOWNMEMBER(<set1>, <set2>|ALL [,RECURSIVE])
  This TM1 -specific MDX function is similar to the DRILLDOWNMEMBER function from Microsoft , but it has been adjusted to match the functionality of the Expand button {bmct expand_button.bmp} on the Subset Editor.
- TM1Member
  This function returns a member from a specified tuple.
- TM1SORT(<set>, ASC|DESC)
  This TM1 -specific MDX function sorts <*set*> alphabetically.
- TM1SORTBYINDEX(<set>, ASC|DESC)
  This TM1 -specific MDX function sorts <*set*> by the index value of the members.
- TM1SUBSETALL(<dimname>)
  This TM1 -specific MDX function returns the TM1 subset All of <*dimname*>.
- TM1SubsetToSet
  This function returns the members of a TM1 subset.
- TM1TupleSize
  This function returns the number of members in a tuple.
- TM1 -Specific MDX expressions
  TM1 supports several TM1 -specific MDX expressions. You can apply these expressions while developing MDX applications to run against the server or

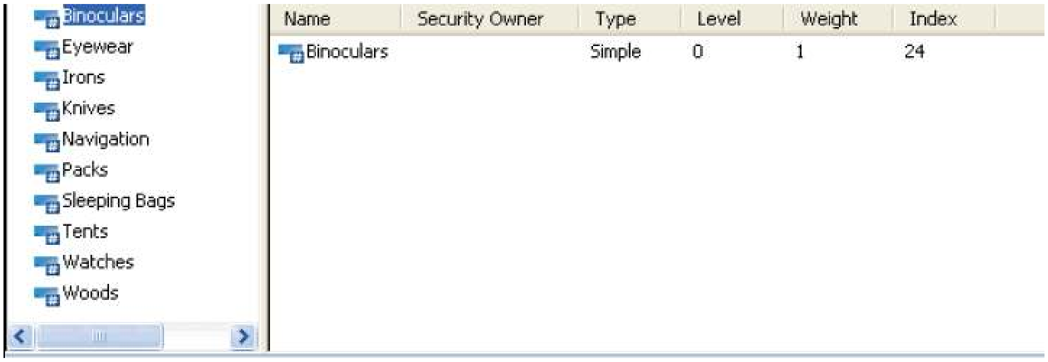
when creating/editing dynamic subsets in the Expression Window of the Subset Editor.

- <dimension>.<subsetname>
  This TM1 -specific MDX expression returns members of <subsetname> in <dimension>.
- <member>.ANCESTORS
  This TM1 -specific MDX expression returns the ancestors of <member>.

Example of MDX Function:



```
{TM1SORT( { TOPCOUNT( {TM1FILTERBYLEVEL( {TM1SUBSETALL( [Products] )}, 0)}, 10.000000,
[Sales Plan].([Subsidiaries].[TOTAL COMPANY],[Channels].[Department Store],[Months].[Total
Year],[Versions].[Budget Version 1],[Sales Plan Measures].[NET SALES REVENUE])) }, ASC)}
```



Binoculars
Eyewear
Irons
Knives
Navigation
Packs
Sleeping Bags
Tents
Watches
Woods

Name	Security Owner	Type	Level	Weight	Index
Binoculars		Simple	0	1	24

Figure 8 - shows the mdx formula

Hint:

- MDX – test it out in the Subset Editor … if it doesn't work there it won't work on your sheet. If passing parameters … make sure " are quoted if the quote needs to be in the statement passed in, i.e.:

  To get this:

  {TM1FILTERBYPATTERN({TM1SUBSETALL([model])}, "*2.5*")}

Need :

="{TM1FILTERBYPATTERN({TM1SUBSETALL([model])}, ""*" & T19 & "*"")}"

- In the Subset Editor, if you view the expression window, you will often see a reference to Subset Basis.
- This statement will not always work when used in a MDX statement in Excel and does not work on the web. For instance you might see the following in the Subset editor when searching for a pattern : TM1FILTERBYPATTERN(TM1SubsetBasis(), 'ABC*')". This MDX will fail on the web. Replace TM1Subsetbasis with an explicit reference to the set you want to use, i.e.

TM1FILTERBYPATTERN([dim name].[subset name], 'ABC*') or
TM1FILTERBYPATTERN({TM1SUBSETALL([dim name])}, 'ABC*')

Useful MDX Statements

- Parent item and it's Children:

="{ [model].["&F12&"], [model].[" & F12 & "].children }"

- Level:

="{TM1FILTERBYLEVEL({TM1SUBSETALL([model])}," & F13 &")}"

- Filter:

="{FILTER ((TM1FILTERBYLEVEL({TM1SUBSETALL
([model])}," & F13 & ")), ([SalesCube].[Month].[Year]) > " & F14&")}"

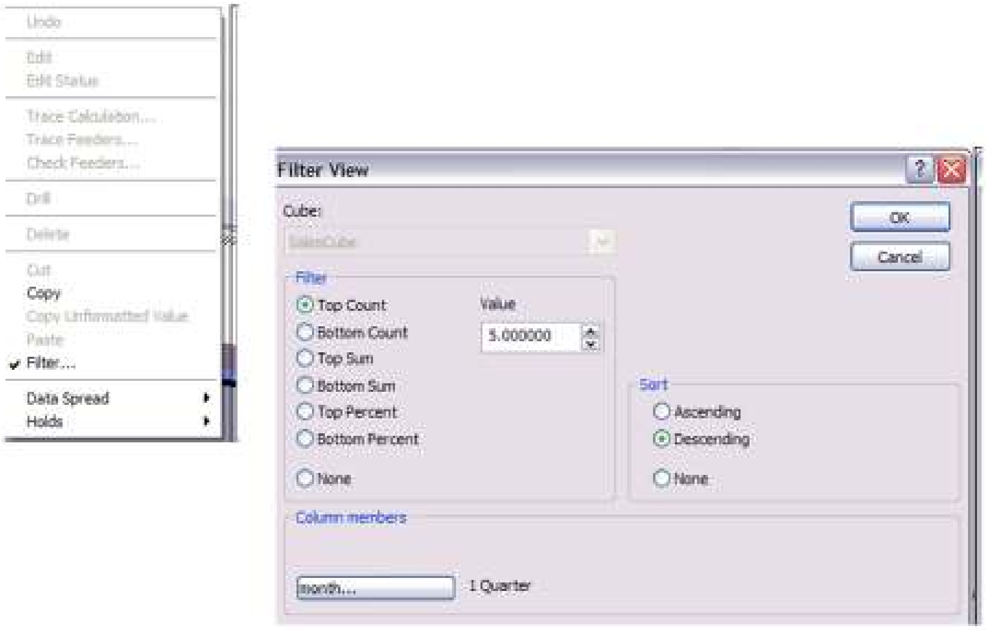
Source:
http://pic.dhe.ibm.com/infocenter/ctm1/v10r1m0/index.jsp?topic=%2Fcom.ibm.swg.ba.cognos.tm1_ref.10.1.0.doc%2Fc_tm1-specificmdxfunctions_n20311.html

Formatting Tips

- Copy data from your form to the format rows so you can see what formatting you are applying. Be sure to copy as values.
- To add new columns … insert prior to the last column. This way Excel will automatically adjust your format range for you. If not, you can adjust in the Insert <> Name dialog by editing the TM1RPTFMTRNG name.

Filtered Reports

- Start with a filtered view (right click on any view)
- Additional formula added to sheet: =TM1RPTFILTER($B$9,"[month].[1 Quarter]","TOPCOUNT",5,"desc")



Cube: SalesCube

● Top Count

Value 5.000000

● Descending

month... 1 Quarter

Figure 9 – shows the filter view

Cell Reference

- As of 9.4.1 FP1
  - SUBNM will open on double click in Excel even if has cell references
  - Action button ability to use cell reference for DB name

Active Form Tips

- Don't use name ranges for long lists … use a named subset or MDX so list does not have to be sent to server to generate view.
- When adding data from another cube … do not point it's DBRW formula back to the TM1RPTView statement
- When adding new data from the same cube … you'll need to rebuild to update the view and retrieve the data
- Zero suppression will be done on the data in the view – behaviour should be the same as if you look at the same data in a cube view

Cell Protection

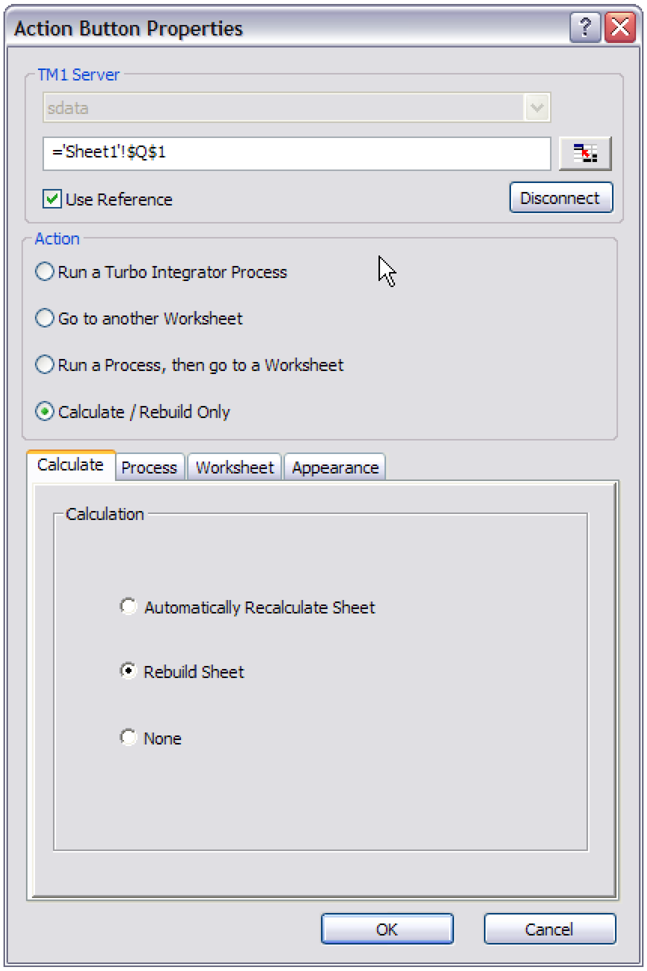
- Will not work with passwords
- Apply lock setting to the format area for any rows in the active part of the report. Will get copied down (if apply directly to the report area, will get reset on recalc)
- Useful if want to prevent users from being able to input into certain areas

Performance

- Don't use named range with long lists
- Watch out for extra columns when you publish on the web … delete and re-save before publishing
- Delete any hidden sheets not being used

Action Button

- Ability to reference the server name (no more manual editing every time you change the server name)
- Calculate/Rebuild Only option
- Calculate Tab (pre any further actions)
- Calculate option on Process and Worksheet page



Action Button Properties

TM1 Server

sdata

='Sheet1'!$Q$1

☑ Use Reference

Action

○ Run a Turbo Integrator Process

○ Go to another Worksheet

○ Run a Process, then go to a Worksheet

● Calculate / Rebuild Only

Calculation

○ Automatically Recalculate Sheet

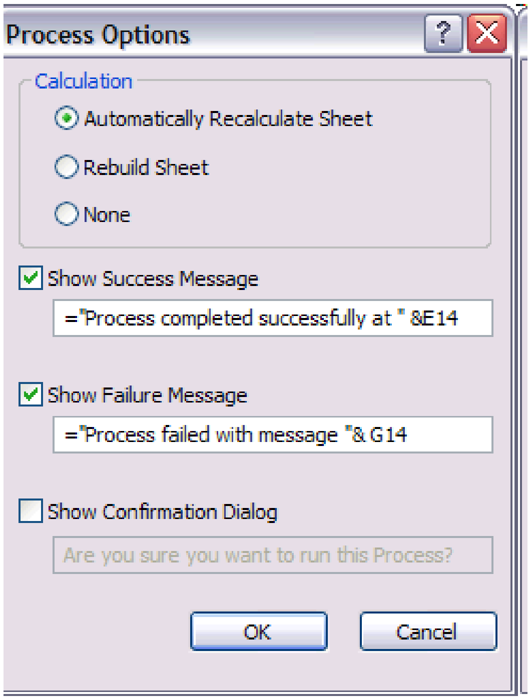
● Rebuild Sheet

○ None

Figure 10 – shows the action button

Add Custom Messages

- Can add custom error messages into the Dialogs available when running a TI process
- Set field to: = cell reference



Process Options

Calculation

● Automatically Recalculate Sheet

○ Rebuild Sheet

○ None

☑ Show Success Message

="Process completed successfully at " &E14

☑ Show Failure Message

="Process failed with message "& G14

□ Show Confirmation Dialog

Are you sure you want to run this Process?

Figure 11 – shows the adding of a custom message

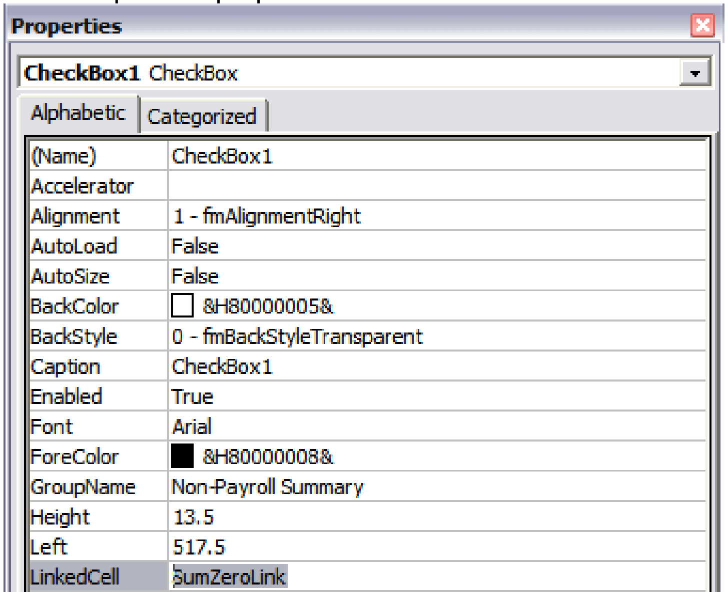
Example 1 – Building Active Report – an option to adding zero suppression to an active form

Steps to add zero suppression to an active form

The below steps are an option to help automate the process of zero suppression within an active forms

1. Add a check box

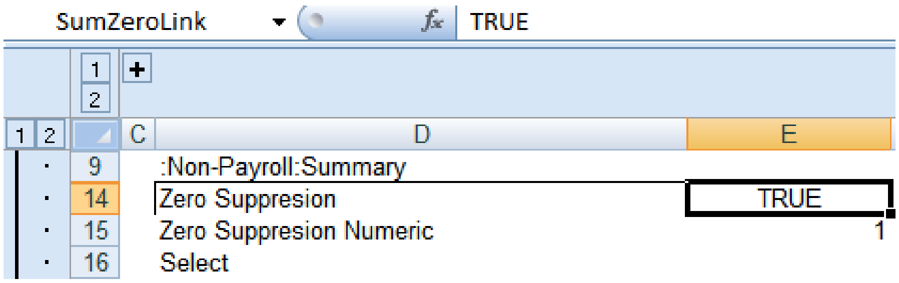
2. Open the properties and add the cell to link the value of the check box



(Name)	CheckBox1
Accelerator	
Alignment	1 - fmAlignmentRight
AutoLoad	False
AutoSize	False
BackColor	□ &H80000005&
BackStyle	0 - fmBackStyleTransparent
Caption	CheckBox1
Enabled	True
Font	Arial
ForeColor	■ &H80000008&
GroupName	Non-Payroll Summary
Height	13.5
Left	517.5
LinkedCell	SumZeroLink

Figure 12 – shows the check box

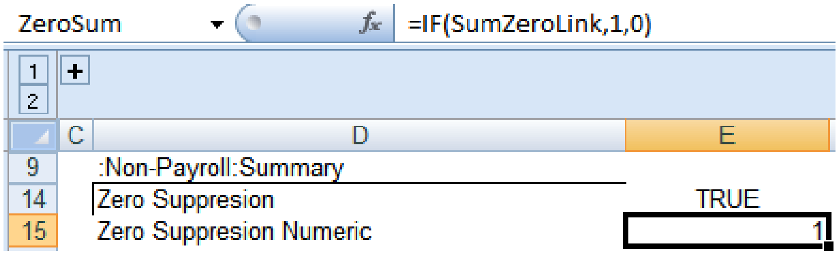
3. In this example users will name the cell E14 with the name SumZeroLink and after that link the checkbox value.



	D	E
9	:Non-Payroll:Summary	
14	Zero Suppresion	TRUE
15	Zero Suppresion Numeric	1
16	Select	

Figure 13 – shows zero suppression set to "TRUE"

4. Next the user will transform the value of the cell into 1 or 0. =IF(SumZeroLink,1,0) And name this cell in this case as "ZeroSum".



ZeroSum =IF(SumZeroLink,1,0)

	C	D	E
9		:Non-Payroll:Summary	
14		Zero Suppresion	TRUE
15		Zero Suppresion Numeric	1

Figure 14 – shows the numeric setting as "1"

5. Finally we will assign the ZeroSum name range in to the TM1RPRTVIEW formula. =TM1RPTVIEW(Server&":Non-Payroll:Summary", ZeroSum,)